STEP and OMG Product Data Management Specifications

A Guide for Decision Makers

Submitted by:

Representatives from the OMG and STEP Standards Communities

Submission Contacts

Owner / Editor: David Starzyk
Address: Boeing Aircraft and Missiles Systems
PO Box 516
St. Louis, MO. 63166
USA

Telephone: (314) 233-6134
FAX: (314) 777-2928
E-Mail: david.t.starzyk@boeing.com

Primary STEP Contact: David Price
Address: IBM Corporation
5300 International Blvd.
North Charleston, SC. 29418
USA

Telephone: (843) 760-4341
FAX: (843) 760-3349
E-Mail: dmprice@us.ibm.com

Primary OMG Contact: Larry Johnson
The MacNeal-Schwendler Corp.
2975 Redhill Avenue
Costa Mesa, CA. 92626
USA

(714) 540-8900
(714) 979-2990
larry.johnson@macsch.com

Additional Submission Contacts

David Briggs	The Boeing Company
Ed Stanton	MSC Software
Larry McKee	IBM Corporation
Jim Kindrick	ERIM
Dr. Mario Leber	ProSTEP GmbH
Dr. Lutz Laemmer	ProSTEP GmbH
Rob Howard	EuroSTEP Limited
Rogerio Barra	ATI
Ed Barkmeyer	NIST
John Jernigan	Enovia
Steve Waterbury	NASA
Rick Bsharah	Ford
Duane Silkworth	Metaphase/SDRC
Mangesh Bhandarkar	Sherpa

Executive Summary

Product Data Management (PDM) is an increasingly important technology for engineering and manufacturing enterprises. The information technology that supports Product Data Management requires standards in order to allow interoperability between systems and the sharing of product information between organizations. Two important sources of standards in this area are the Object Management Group (OMG) and the International Organization for Standardization (ISO) STEP community (officially ISO TC184/SC4 Industrial Data). This document will show how these two standards work together to support engineering and manufacturing processes in today's business environment.

OMG and ISO are organizations that develop and promote standards. The OMG and STEP communities work independently to develop standards for different purposes and that differ in scope, abstraction level, and operational characteristics. The OMG standard defines a standard interface to the services of a PDM system in a distributed, object oriented environment. The STEP standard defines a standard representation of product data including data that is typically managed by a PDM system.

However, the OMG and STEP communities have worked together to harmonize their respective standards in the area of Product Data Management Standards so that the standards complement each other rather than conflict or overlap. A formal liaison relationship has been established between OMG and ISO TC184/SC4 to facilitate this harmonization. This document provides a technical comparison of the OMG and STEP standards; a high level architectural approach for the use of these complementary standards to satisfy business requirements in real world scenarios; and guidelines for how and when to use these standards.

Table of Contents (A Guide for Decision Makers)

Table of Figures

1 Introduction

The focus of this document is to provide information on the comparison and harmonization of the Product Data Management Standards developed and promoted by both the OMG and STEP Standards Communities. The OMG and STEP communities have come together to specify the complementary aspects of these two standards defined herein. This document outlines scenarios and defines the high level architectural approach for the use of these complementary standards and provides decision criteria for how and when to use these standards to satisfy business requirements.

STEP and the PDM enablers are different in purpose, scope, abstraction level, and operational characteristics. Despite these differences, and because of these differences, each provides an essential piece in the process of defining and managing product data.

1.1 Intended Audience

This document was written for managers, designers and implementers of applications, which will employ OMG and STEP Standards to share and exchange PDM information.

2 The Organizations

2.1 The Object Management Group (OMG)

The Object Management Group (OMG) is an open consortium of software vendors and users dedicated to creating and popularizing object-oriented standards for application integration, based on existing technology. The organization's charter includes the establishment of detailed service and interface specifications that provide a common framework for the development of component applications for distributed object computing systems. Conformance to these specifications allows independently developed off-the-shelf software packages to participate in distributed systems designed by end-users in a heterogeneous computing environment that includes all major hardware/software platforms. OMG specifications are used worldwide for distributed applications in Manufacturing, Finance, Telecommunications, Electronic Commerce, Health Care, Public Utilities and Life Sciences Research.

OMG defines *object technology* as software that models the real world through "objects." These objects are the exposed representation of the attributes, relationships and methods provided by underlying software components. *Object management* is the design, deployment, operation and maintenance of systems based on object technology. A key benefit of an object-oriented system is its ability to expand in functionality by replacing existing components and extending the set of components (the "object instances") that provide a given interface, without altering other components of the system. Object management results in faster application deployment, easier maintenance, enormous scalability and reusable software.

In 1996, The OMG Manufacturing Domain Task Force, one of the standards development bodies of the OMG, released a Request for Proposal for a standard set of object interfaces for PDM services. Seven major PDM vendors responded with separate "initial proposals," each of which were reviewed and commented on by the Task Force. All submitting vendors formed a Joint Product Data Management (JPDM) submission team with the objective of authoring a single consensus submission. Facilitated by the JPDM chairman from the National Center for Manufacturing Sciences through MSC Software Corporation, these vendors (Enovia, Fujitsu, Matrix-One, Metaphase/SDRC, Sherpa, Digital and NIIIP) worked together to develop a single joint "revised proposal". This proposal was extensively reviewed by the Task Force, including many major manufacturers and other PDM vendors, and further revised to meet their concerns. The resulting final specification was formally adopted by vote of the OMG Domain Technology Committee and the OMG Board of Directors in August 1998 as the "PDM Enablers" interface standard.

2.2 The ISO Subcommittee on Industrial Data

STEP, the Standard for the Exchange of Product Data, is an activity producing International Standards under the auspices of ISO TC184/SC4. SC4, named Industrial Data, is a standing subcommittee of the International Organization for Standardization (ISO). TC184, named Industrial automation systems and integration, is a Technical Committee of ISO. SC4 is developing standards that provide capabilities to describe and manage product data throughout the life of the product. SC4 is made up of Working Groups and the Quality Committee. STEP is formally ISO 10303 - Industrial automation systems and integration - Product data representation and exchange, and is broken into a series of parts. One of these series of parts is Application Protocols (APs). An AP defines an information model for a specified scope. SC4 produces other industrial data standards as well.

The STEP Standards Development

The development of STEP as an international standard is driven in large part by industry and country consortia. Two of the larger of these consortia, PDES, Inc. and ProSTEP, are the major authors of the PDM Schema. Each of these consortia has multiple APs under development. PDES, Inc. member companies are the authors of AP203 Configuration controlled 3D design of mechanical parts and assemblies, AP232 Technical data packages, AP210 Electronic assembly, interconnect and packaging design, among others. ProSTEP members are the authors of AP214 Core data for automotive mechanical design processes and

AP212 Electrotechnical design and installation. PDES Inc and ProSTEP are working together to harmonize these APs in their overlapping PDM domain. The PDM Schema is the result of this effort.

The deployment of STEP into industry is also facilitated by the industry and country consortia that support testbeds for vendor software testing, coordinate priorities for new capability industry requests from vendors and who support industrial pilot projects within and between their members. AP203, for example, has been successfully piloted and has been in production use in industry since 1995. AP214, AP232, other APs, and the STEP PDM Schema are also the basis for industrial pilot projects. One pilot using the STEP PDM Schema is expected to lead to its production use during 1999.

3 The Standards

3.1 Intent of PDM Enablers

3.1.1 The Object Management Architecture

The Object Management Architecture (OMA) lays out the fundamental principles for distributed systems interfaces as defined by the Object Management Group.

The core idea of the OMA is that an *object* is a collection of software that maintains useful knowledge or performs useful functions and is wholly *encapsulated within* a component of a distributed system. The knowledge and functions of that object are exposed to the rest of the distributed system only through *operations* on the object.

An *operation* is a function that is performed by a software implementation (called the *server*) within the component that contains the object. The function execution begins with an invocation by some component (called the *client*), that identifies the object and the operation and passes some set of input data units to the server. The execution ends with a termination within the server and a return to the client passing some set of output data units.

Unlike objects, data units are actually transmitted between components. Every data unit has a unique *data type*, which characterizes the nature of its usage. For practical purposes, a data type is characterized as a set of possible values. There is a special data type, called an *object reference*, whose values are the identifiers of encapsulated objects in the distributed system.

An operation is said to have a *signature*, a *definition*, and one or more implementations. The signature specifies the name of the operation, the types of the data units that are passed on invocation, and the types of the data units that are returned on termination. The definition specifies what the operation does, as perceived by any component of the distributed system other than the object implementation. An implementation of the operation on a given object is the set of software that performs the defined function on that object according to the specifications.

The collection of all operation signatures and definitions that are supported by an object is said to be its *interface*. The interface to an object is its complete external characterization within the distributed system. More than one object can have the same interface, and thus the interface is a means of typing or classifying objects. (And their interface types can categorize data units that are object references.)

The components in a distributed system can then be characterized by the objects they maintain and the interfaces those objects exhibit, and all interactions of the components in the system can be described as invocations of operations on those objects.

It is worth observing that this same model is used under different names in the Java community, the Microsoft DCOM community, and parts of the World-Wide Web community, as well as OMG.

3.1.2 The Common Object Request Broker Architecture (CORBA)

The Common Object Request Broker Architecture is a refinement of the OMA, with the following additional ideas:

A component of a distributed system can be constructed independently of all other components, so long the object interfaces it provides are well-defined, the (external) object interfaces it uses are well-defined, and there is a standard means for performing the invocations over the network.

The object interface can be defined in a language that is independent of the component implementation languages, as long as there is a standard mapping from the specification language to an implementation of the invocation in the programming languages of the components. There are actually two mappings -- one at the "client", who performs the invocation, and one at the "server", which implements the object and the operation. These can be in different languages on different platforms, as long as both map to the standard for conveying the invocation over the network.

The function of the Object Request Broker is to perform the dynamic linking operation that connects the client that invokes an operation on a given object to the server that implements that operation on that object.

In practice, a CORBA package contains libraries that implement the broker functions and the standard invocation conventions over the network, and a compiler for the standard Interface Definition Language (ISO 14750) that generates interface-specific bindings of the communications libraries to specific programming languages. In effect, the client programmer sees a library that implements the remote object interfaces as subroutine calls, and the server programmer sees a main program calling his operation implementations as subroutines. And the IDL specification defines the application programming interface (API) to those "subroutines".

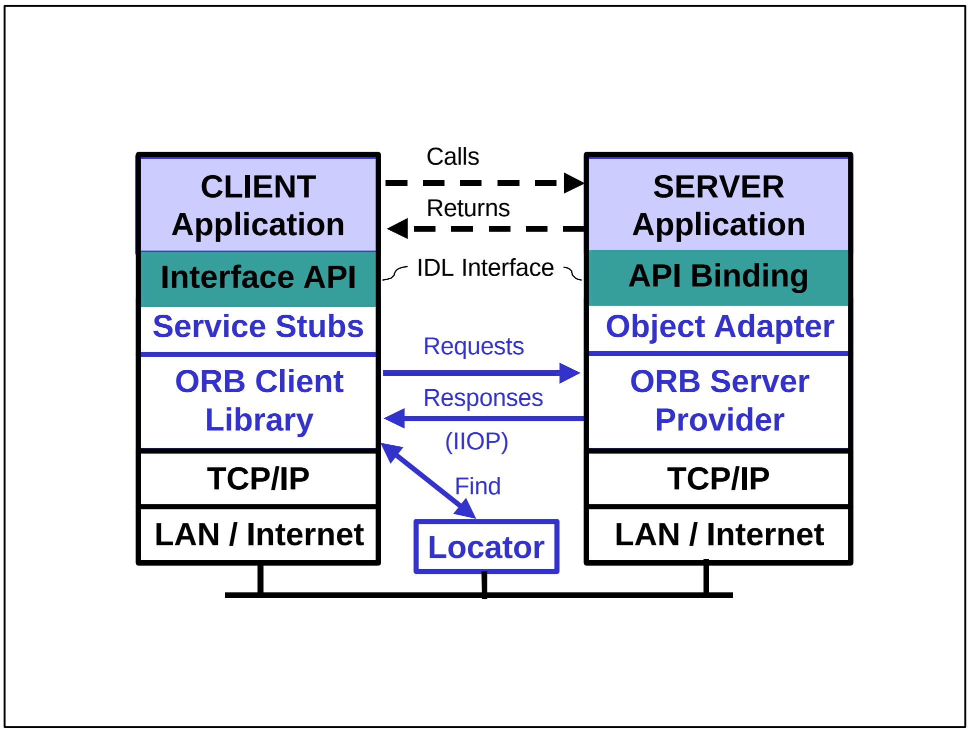


Figure 1 - The CORBA Software Architecture

3.1.3 The PDM Enablers

The PDM Enablers are a standards-based Application Programming Interface (API), specified in IDL, that makes PDM services available in a CORBA environment to other systems that require them (such as CAD, CAE, CAM systems mentioned above, and even other PDM systems). Following the CORBA model, CAx systems can use the PDM Enablers API and the standard network interfaces to interact directly with any conforming PDM system.

The PDM Enablers provide direct interfaces to support Document management, Product Structure management, Change management, (Options) Configuration management, and Manufacturing implementation specifications, and include support for Views, Effectivities, and Baselines. A special Enabler provides for import and export of STEP exchange files.

The PDM Enablers are intended to provide interface to the services of a PDM system operating in an arbitrary manufacturing enterprise with an enterprise-specific schema. The Enablers constitute a model of PDM interfaces that can be mapped to commercially available PDM systems. As such, they are not meant to be a definitive model of PDM Objects or Systems. They are intended to serve Work-in-Process PDM systems as well as Enterprise-level Release Management PDM Systems, and they are intended to support most engineering activities, but not all PDM activities. Interfaces for PDM administration, for example, are not in the scope of the standard.

The PDM Enablers interface model is sparse in explicit attributes. It is, however, able to support rich attribution consistent with the system that actually provides these interfaces. This allows the interfaces to be used in a wide variety of contexts, because they are able to support detailed attributes from various standards, or attribution particular to a specific enterprise.

The conspicuous elements of the PDM Enablers' model are the inheritance hierarchy and the relationships among objects. The STEP PDM Schema was used as a reference model in constructing this framework of relationships, and the Enablers adhere to that model, except in a few cases. Where the STEP PDM Schema did not cover an area of functionality, a specific STEP Application Protocol was used to the extent possible. For example, functional units of STEP AP214 were used as the reference models for the Manufacturing Implementation and Configuration Management Enablers.

Most of the differences between the Enablers and the STEP standards arise from the need to support work-in-process PDM services and other dynamic services of the PDM system, and from the need to support enterprise-specific extensions and modifications to the PDM schema.

3.2 Intent of STEP PDM Schema

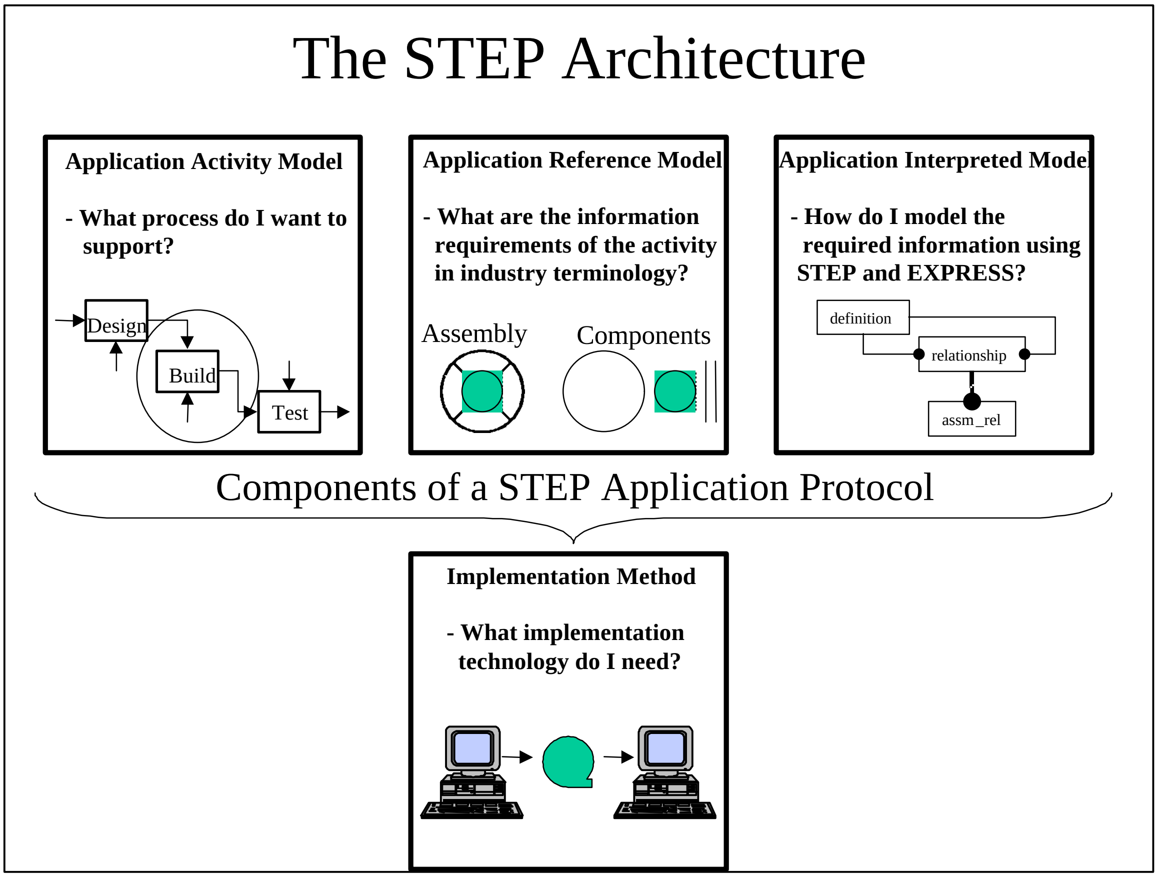


Figure 2 - STEP Application Protocol Architecture

The STEP Standards

As was stated earlier, the information models that comprise the STEP standards are known as Application Protocols (APs). An AP consists of the following major elements:

- an activity model describing the business process the information model supports,
- an application reference model specifying the information requirements and
- a schema for data structures, called an application interpreted model (AIM), which is the basis for implementations of the standard.

STEP standards developers have made a conscious decision to focus on the information required by industrial processes rather than the processes themselves as these are seen to change over time while the underlying information requirements are longer lasting. This focus on information allows STEP to support data exchange, some forms of data sharing as well as long term data retention. To that end, the STEP community has created a data specification language, called EXPRESS, as well as implementation methods for file exchange and a data access interface based on that language. The AIM of an AP is specified using

EXPRESS and a conforming implementation must utilize the AIM schema in combination of one of the implementation methods.

The PDM Schema

The STEP PDM Schema is a harmonized data model capable of supporting a central, common subset of the data typically managed within a PDM system. Its scope was developed as the harmonized **intersection** of requirements and data structures from a range of STEP Application Protocols all generally within the domains of design and development of discrete electro/mechanical parts and assemblies. Based on this scope, not all functionality found in typical PDM systems, the relevant STEP Application Protocols or the OMG PDM Enablers is covered.

The intent of the STEP PDM Schema is to provide, from the STEP community, a single schema along with consistent mappings into that schema supporting core PDM capabilities. This enables PDM vendors to implement STEP-based PDM capability once, based on the PDM schema, and have that development be reusable in, or at least interoperable with, implementations based on the relevant STEP Application Protocols. A secondary benefit of the STEP PDM Schema is that it enables harmonization with other PDM-related standards, such as the OMG PDM Enablers, for large segments of industry. A brief overview of the STEP PDM Schema capabilities follows:

- By definition, a Product Data Management (PDM) system is something that manages data about products. Necessarily then, at the core of PDM information is product identification. A product in STEP represents the concept of a general, managed item within a PDM system. In the STEP PDM schema, the general product concept may be interpreted as either a Part or a Document. In this way, parts and documents are managed in a consistent and parallel fashion.

- Also central to the functionality of many PDM systems is identification of external files (both digital and physical), their relationship to managed documents, and their reference to core product identification.

- Classification of products is important in a PDM system for information classification and retrieval: it also supports basic type distinction between products that are parts and those that are documents

- Product properties are integrally related to the definition of an identified product, and so are naturally also included in the central core PDM information.

- Various general authorization and organizational data that are related to core product identification plays an important role in PDM systems.

- Product structures are the principal relationships that define assemblies and product configurations. Both part and document structures are addressed in the STEP PDM schema. Configuration identification and effectivity information related to these structures is also included.

- Finally, structures to manage the documentation of requests and corresponding orders for engineering action, in support of the change management process are included. Also included are representations for contract and project identification.

4 Relationships Between the Standards

4.1 The Roles of Separate Specifications

As stated in other parts of this document, the OMG PDM Enablers and the STEP PDM Schema are complementary specifications that work together. They can be viewed as separate necessary documents that concentrate on different aspects of a single unified specification of a standard system of interoperable PDM clients and servers.

The Reference MODEL for Open Distributed Processing (RM-ODP) provides a conceptual framework in which each of these specifications plays a separate well-defined role.

4.1.1 Introduction to RM-ODP Viewpoints

The Reference MODEL for Open Distributed Processing (RM-ODP) is an adopted international standard specification (ISO/IEC 10746). It is independent of the Object Management Group or CORBA, but it is widely accepted by the OMG as a conceptual framework to which OMG specifications should relate. The document **ISO/IEC 10746-1 ODP Reference Model Part 1** presents an overview of RM-ODP.

Among other important concepts, RM-ODP shows how a total design is separated into areas of concern identified by five (5) separate *viewpoints*, each of which satisfies the requirements of a separate group in the design process. These viewpoints are named Enterprise, Information, Computational, Engineering, and Technology. RM-ODP considers this set of viewpoints to be simple but complete enough to cover all the domains of architectural design needed for open distributed object processing.

The viewpoints are projections onto certain sets of concerns rather than independent layers or design methods. Specifications of a system from each viewpoint are partial views of the complete system using a viewpoint-specific specification language. There are consistency constraints between the different viewpoint specifications that arise from the relationships between the real word entities represented in the different specifications and from the fact that the same entities may be represented in more than one specification.

4.1.2 Viewpoints for PDM Enablers and STEP

To be clear, neither the STEP specifications nor the PDM Enablers specification were designed from the start to be formally compatible with the RM-ODP framework, and the STEP standards are not intended to specify object-oriented distributed processing, so they fit inexactly into the RM-ODP framework. The correspondence with RM-DOP is somewhat serendipitous and inexact. Nevertheless, RM-ODP illustrates the intent of these specifications very naturally.

The STEP PDM Schema is a specification primarily in the RM-ODP Information Viewpoint. The OMG PDM Enablers is a specification primarily in the RM-ODP Computational Viewpoint. They neatly fill undefined viewpoints in each other's framework. Table 1 - OMG and STEP Specifications within RM-ODP , and the following discussion explains each viewpoint and shows how various OMG and STEP specifications fit into the RM-ODP framework.

RM-ODP Viewpoint	**OMG PDM Enabler Specifications**	**STEP Specifications**
Enterprise	Mfg. White Paper, PDM Enablers RFP	Application Activity Model (AAM)
Information	**by Customer, or STEP**	Application Reference Model (ARM) Application Interpreted Model (AIM) (PDM Schema)
Computational	PDM Enablers	**Out of Scope**
Engineering	CORBA, OMA IDL language mappings	STEP file exchange structure STEP data access interface and C, C++, Java or OMG IDL mapping
Technology	by PDM Enablers Vendors	by STEP Processor Vendors

Table 1 - OMG and STEP Specifications within RM-ODP

4.1.2.1 Enterprise Viewpoint

The enterprise viewpoint is concerned with the business activities of the specified system. An enterprise specification is a model of the system and the environment with which the system interacts. It covers the *role* of the system in the business, and the human user roles and business policies related to the system.

- *Enterprise Viewpoint for the OMG PDM Enablers*

  Non-normative information in the enterprise viewpoint has been established in the work of the Manufacturing Domain Task Force (MfgDTF). A Manufacturing Enterprise Systems white paper presents the rationale for the work of the MfgDTF well as a high level Manufacturing Enterprise Model. The PDM Enablers Request for Proposal (RFP) presents the part of this high level model concerned with the product development process. Section 1.12 of the PDM Enablers specification is a mapping between the enterprise viewpoint of the RFP and the computational viewpoint of the PDM Enablers specification.

- *Enterprise Viewpoint for the STEP PDM Schema*

  The Application Activity Model (AAM) in a STEP Application Protocol is a specification in the enterprise viewpoint. It provides an informative statement of scope by specifying the processes, information flows, and functional requirements of an application. The AAM of a STEP standard is often specified in the IDEF0 modeling language.

4.1.2.2 Information Viewpoint

The information viewpoint focuses on the semantics of the shared information communicated and processed. It is represented in terms of information objects and their relationships, where information objects are abstractions of entities that occur in the real world, in the ODP system, or in other viewpoints. An information specification may also specify constraints, transformations, and information flows.

- *Information Viewpoint for the OMG PDM Enablers*

  The PDM Enablers deliberately leaves the information viewpoint open to be defined by the PDM vendors and individual sites in their default and customized schemas.

  The STEP PDM Schema defines an appropriate PDM model in the information viewpoint. In fact, the STEP PDM Schema was used during the development of the OMG PDM Enablers specification, and is the primary reference information model for the PDM Enablers.

- *Information Viewpoint for the STEP PDM Schema*

  The STEP Application Reference Model (ARM) is an information specification that defines requirements and constraints using industry terminology. These are the normative requirements that implementations of the AP must meet. It has characteristics of both the information and enterprise viewpoint. It is specified using English to describe a set of application objects and relationships between those application objects. An informative graphical representation of the ARM may be specified in IDEF1X and/or EXPRESS.

  The STEP Application Interpreted Model (AIM) is clearly in the information viewpoint. It defines specific data elements, relationships, and constraints of a STEP implementation schema, and is specified using EXPRESS. A specification of the relationship between the ARM and AIM is specified in a mapping table within the AP constraining populations of the AIM data structures. The STEP PDM Schema is the harmonization and intersection of several AIMs.

4.1.2.3 Computational Viewpoint

The computational viewpoint is concerned with the description of the system as a set of software components (*objects*) that interact at *interfaces* - enabling system distribution. A specification in the computational viewpoint may include the functions provided by the components, and their relationship to the functions of the distributed system, the interface at which a function is invoked/activated, and the protocols governing the sequence of actions and events at interfaces.

- *Computational Viewpoint for the OMG PDM Enablers*

  The PDM Enablers is a specification in the computational viewpoint. It specifies a set of objects that interact at interfaces. It is documented with the help of the Unified Modeling Language™(UML) and is specified in OMG Interface Definition Language ™ (IDL). All OMG IDL specifications are in the computational viewpoint.

  The PDM Enablers, as defined in UML, may be an appropriate computational model with a variety of engineering and technology environments, including non-CORBA environments.

- *Computational Viewpoint for the STEP PDM Schema*

A computational specification is a set of object interfaces established to enable distribution. Since STEP is not concerned with the problem of distributed processing, and does not specify object-oriented operations, the computational model is out of scope of the STEP standards.

An argument can be made that the STEP Data Access Interface (SDAI) in conjunction with a particular EXPRESS schema defines a computational model. However, the SDAI mechanism is oriented to element-level read-write operations directly against the information schema, not encapsulated object distributed processing.

4.1.2.4 Engineering Viewpoint

The engineering viewpoint is concerned with the mechanisms implementing the distributed system. An engineering specification defines a networked computing infrastructure that supports the system structure defined in the computational specification and provides the distribution transparencies that it identifies. It describes mechanisms corresponding to the elements of the programming model, effectively defining an abstract machine, which can carry out the computational actions, and the provision of the various transparencies needed to support distribution.

- *Engineering Viewpoint for the OMG PDM Enablers*

  The engineering viewpoint used by the PDM Enablers is the Common Object request Broker Architecture™ (CORBA™) and the Object Management Architecture™ (OMA™). CORBA defines the interoperable mechanisms that enable distributed processing, and the OMA defines the separation and interworking of software using these mechanisms. Various language mappings define how the OMG Interface Definition Language ™ (IDL) is expressed in specific programming languages.

- *Engineering Viewpoint for the STEP PDM Schema*

  The engineering viewpoint is established by the STEP file exchange structure, which defines the syntax of data in an exchange file under the definition provided by an EXPRESS information model.

  SDAI and its programming language mappings are also in the engineering viewpoint.

4.1.2.5 Technology Viewpoint

The technology viewpoint is concerned with the detail of the components from which the distributed system is constructed. A technology specification defines how a system is structured in terms of hardware and software components.

- *Technology Viewpoint for the OMG PDM Enablers*

  The technology viewpoint for the PDM Enablers is established by the vendors of PDM Enablers implementations, and by the specific implementation and deployment of PDM systems at an enterprise's site.

- *Technology Viewpoint for the STEP PDM Schema*

  The technology viewpoint for the STEP PDM Schema is established by the vendors of STEP processors and by the specific deployment at an enterprise's site.

4.2 Complementary Aspects of the two Standards

4.2.1 Purpose

The purpose of the OMG PDM Enablers is to provide access to the services of product data management systems from various application software systems in a manufacturing enterprise. Activities supported by such “client” applications encompass product conception and planning, product design, manufacturing engineering, production, delivery and maintenance. The emphasis is on providing interfaces for the management of product data:

- Store (check in) and retrieve (check out) product “documents” (models, drawings, data sets, etc.)
- Track assembly/component relationships, alternate and substitute parts, provide bills of materials and perform “where used” and “where useable” searches
- Manage multiple versions and their effectivities.
- Control update and release.
- Provide overviews of the relationships among products, product specifications, process specifications, maintenance specifications, and the engineering and management tasks that create, use and modify them.

The PDM Enablers are intended to support manufacturing activities, and particularly engineering activities, operating on specifications in varying stages of completion and release. For this reason the emphasis is on the services of the PDM and other engineering data management systems that implement these interfaces.

The purpose of the PDM Schema and the related STEP standards (ISO 10303 Parts 203, 207, 208, 209, 210, 212, and 214), also known as APs, is to provide for the exchange of one or more sets of product specifications between software systems. The emphasis is on providing mutually comprehensible data:

- product models, drawings, data sets, documents, etc.
- assembly/component relationships, alternate and substitute parts, bills of materials
- identification and description of multiple versions and their effectivities.
- release levels and change management documents
- relationships among products, product specifications, process specifications, and maintenance specifications.

The STEP APs are intended to support *controlled release*, as between a prime contractor and a supplier of a part or assembly. The consequence of this view is that the AP model is a static description of product information, rather than services.

This is not to say that there is no significant overlap in the objects and properties modeled, as the above bullets indicate. But there is a difference in approach that does not lead to a simple one-to-one mapping between the objects in the STEP data models and those in the OMG interface specifications.

4.2.2 Domain Scope

The normative units of the STEP standard are application protocols (AP). There are many APs, in varying stages of development and adoption as International Standards. The standard AP commonly in use, and used as a reference in the development of the PDM Enablers, is AP203 – ISO 10303-203:1994, *Configuration-controlled Mechanical Design.* But there are now many Application protocols under development that expand the scope of STEP to other product development disciplines (e.g., electrical, automotive and software) and across the entire product life cycle (from concept through maintenance, service, and retirement). The STEP PDM Schema was recently developed to provide a common reference model for product configuration and management information across all these domains. Because the needs for product management data vary over these domains, the PDM Schema is organized as a set of "modules" that provide definitions of different categories of product management data.

The PDM Enablers was jointly developed by representatives of several major PDM vendors who brought together their joint knowledge of a wide variety of PDM customer applications. But since their PDM products differed in their capabilities, as did their customer demands, the PDM Enablers was also organized as a set of "modules". These "IDL modules" are the units of conformance to OMG formal specifications, and in the case of the PDM Enablers, it is not required that all implementations provide support for all of the modules. (Nor did all the original submitters intend to do so.)

Unsurprisingly, the "modules" of the PDM Schema and the "modules" of the PDM Enablers tend to parallel one another, consistent with meaningful groupings of PDM information and services. Table 2 - Relationships between STEP PDM Schema 1.1 and OMG PDM Enablers Modules, shows the relationships between the modules, and indicates the strong agreement on the areas of PDM data and PDM services for which standards are wanted.

STEP PDM Schema 1.1	OMG PDM Enablers
Part Structure and Part Properties	PdmProductStructureDefinition
Document Management	PdmDocumentManagement
Alias Identification	in PdmFoundation
Authorization	PdmResponsibility
Effectivity and Configuration	PdmEffectivity, PdmViews
Change Management	PdmChangeManagement
Part Classification	out of scope
out of scope for version 1	PdmConfigurationManagement
out of scope	PdmManufacturingImplementation
(implicit in exchange data set)	PdmBaseline
Not Applicable	PdmSTEP

Table 2 - Relationships between STEP PDM Schema 1.1 and OMG PDM Enablers Modules

PdmConfigurationManagement (product options) and PdmManufacturingImplementation (process specifications) are units of AP214 that are important to several industries but were not included in the scope of any other STEP AP at the time of development of the PDM Schema and therefore not a "common concern" across APs. PdmBaseline services deal with identifying a set of specifications as a conceptual unit, and PdmStep services deal with exchanging a set of specifications as a STEP exchange file. Both of these are covered in other Parts of the STEP standard. Part classification is an area in which the needed services go well beyond the management of product data and are commonly provided by companion products to the PDM systems. So the OMG Task Force explicitly stated that Part Classification concerns were out of scope in the OMG Request for Proposal.

4.2.3 Model Characteristics

As indicated in [Reference Section 4.1 The Roles of the Separate Specifications], the PDM Schema is an information model. It defines the "universe of discourse" for a set of systems sharing product data and product management data – what the engineering objects are and what information describes them. The OMG PDM Enablers is a computational model. It defines the interfaces to a product data management system – what the engineering objects are, how to create them, how to modify them, and how to activate related PDM services. While the engineering objects are almost entirely the same, the other content of the two models is quite different. Each provides a different view of elements of a distributed system supporting engineering and manufacturing operations. And each of these views supports a different aspect of interoperation of PDM systems with other software systems. Both views are necessary to design and implement the "distributed engineering system". And each view is used with a particular "technology" to solve a particular kind of problem in distributed engineering. What is important is that these views are consistent, or can at least be reconciled, in performing operations on product data.

4.2.4 Level of Abstraction

Both the PDM Schema and the PDM Enablers are defined at a level of abstraction that is intended to support broad multidisciplinary activities in product development and to support product information covering the entire product life cycle. Detailed semantics are restricted to product structure, configuration, effectivity, and other product information management concepts. At the level of abstraction of PDM, specific models of product aspects are not of interest. A PDM system captures relevant product descriptions in documents and other product-definition objects that are detailed in other standards.

Thus there is a common level of abstraction and a high degree of overlap in the "engineering objects" described by the two standards.

4.2.5 Standardization vs. extensibility

Generally, for Parts, Documents, and other managed objects and relationships, the OMG PDM Enablers provide a means for finding, setting and examining whatever attributes those objects are given in the PDM schema for a particular enterprise. It standardizes only those attributes that are important to the PDM services themselves. Additional properties and relationships are handled via the Attributable, Qualifiable and PdmNamedRelationship services. This allows the PDM Enablers to be used to support enterprise-specific applications as needed.

The STEP PDM Schema, on the other hand, standardizes a specific set of attributes and relationships that are needed for successful interchanges between trading partners and diverse PDM systems. Therefore, the STEP PDM Schema intentionally does not support additional user-defined attributes. It does support additional relationships, but the intention is that these will be defined by Application Protocols governing particular exchanges.

This is an important difference between the two standards, reflecting the different requirements for their intended use.

4.2.6 Implementation methods

OMG PDM Enablers

The OMG PDM Enablers standard specifies the use of the CORBA standard implementation mechanisms (see 3.1.2 The Common Object Request Broker Architecture (CORBA)). That is, the PDM Enablers specification defines the interfaces to the PDM system in the ISO/OMG Interface Definition Language and requires the PDM vendor to provide an implementation of those services that conforms to the current OMG CORBA/IIOP Specification. That specification also defines standard mappings from the IDL specification to the corresponding APIs in most common programming languages: Ada, C, C++, COBOL, Java, Lisp, and Smalltalk.

The PDM vendor builds a front-end (or “wrapper”) for his PDM services, using a CORBA software package that conforms to the standard. The CORBA package uses the IDL definitions for the PDM Enablers, and provides implementations of the standard CORBA Broker services and networking conventions (the Internet Inter-ORB Protocol -- IIOP) for the PDM Enablers interfaces, bound to “operation invocations” (or function calls) in the PDM vendor’s programming language. The PDM vendor implements these operations/functions using underlying PDM services. In a similar way, any “client” system that wants to use the Enablers to access a PDM system also uses a CORBA-conforming package to create, from the PDM Enablers IDL specifications, a set of “proxy” subroutines for the client’s programming language and operating system. When the client, e.g. a CAx system, and the PDM are installed on the same network and one of these proxy routines is called, the client subroutine uses the standard Broker and network conventions to find and invoke the PDM-vendor-provided implementation of the corresponding PDM Enablers service. This provides “out-of-the-box plug compatibility” based entirely on the PDM Enablers specification and the OMG CORBA standards. (Compatibility of engineering *data*, of course, is the responsibility of the user site.)

STEP PDM Schema

The ISO STEP standard specifies two implementation methods: 1) a text file exchange syntax and 2) a data access interface. The EXPRESS language used to specify the data models in STEP drives both of these implementation methods.

The file exchange syntax, officially the ISO 10303-21 Clear text encoding of the exchange structure, specifies a computer and human readable format for data modeled using EXPRESS. The encoding is straightforward and efficiency issues such as file size are taken into account. Since the PDM Schema is defined in Express, this standard, together with the PDM Schema, defines the form of files that conform to the PDM Schema. This is the simplest and most common form of implementation of the PDM Schema.

The data access interface, officially ISO 10303-22 Standard data access interface (SDAI), specifies an environment and operations for manipulating data based on an EXPRESS data model. There is a related series of ISO 10303 documents that are the programming language specific implementations of the SDAI in C++, C, and Java. The SDAI is described as a data access interface, rather than an application programming interface, because the operations defined do not understand the semantics of the data model. For example, there are operations to set the values of entities with real-valued attributes, e.g. the coordinates of a 3-dimensional point, but no operations to calculate the distance between two points. Therefore, the SDAI capability is at a level similar to the SQL select, update, delete and create operations. More complicated operations may be built from these more atomic operations. As the data model drives the available operations, the data model also drives the granularity of the data access interface.

There is also a standard mapping of SDAI into the ISO/OMG IDL language, allowing SDAI access to STEP EXPRESS data models to be implemented in a CORBA environment. This means that given the STEP PDM Schema, a set of STEP PDM Schema-based IDL specifications are available. These specifications provide a remote data access interface to PDM data. And using the OMG IDL language mappings, they also provide for mapping SDAI to other programming languages.

Relationships

Only the STEP PDM Schema provides a standard form for files of PDM data.

Both standards provide for direct application program access to PDM data, and both provide a standard mechanism for remote access. But there are several differences in purpose and scope between these two sets of specifications:

- The SDAI specifies simple create, delete, select and update methods for manipulating PDM data units; the OMG PDM Enablers specify interfaces for elementary PDM operations that typically manipulate several related PDM information units.

- The SDAI interfaces have no PDM-specific capabilities – they treat the PDM as a database defined by the PDM Schema; the PDM Enablers specify PDM-specific functions, such as check-in/check-out.

- The SDAI specifies access at a low level of granularity, and as such the unit-of-work is controlled by the application program using the SDAI; where the OMG PDM Enablers specify operations on PDM objects, and as such the unit-of-work is controlled by the PDM providing the Enablers interfaces.

In summary, the different capabilities of the STEP PDM Schema and the OMG PDM Enablers lead to decisions about usage. If the application requires exchanges of a large body of consistent PDM data, the PDM Schema file should be used. If the application program needs low level data access (similar to what SQL provides) then the STEP PDM Schema SDAI makes sense. But if the application needs to access true PDM services then the OMG PDM Enablers is the appropriate choice.

4.2.7 Using Both Specifications Together and Separately

To interoperate seamlessly, programs must agree on and use the same specification in all viewpoints.

The greatest level of run-time standardization is gained when the interfaces of the PDM Enablers computational model are used on the OMG CORBA engineering model with clients and servers that both understand the standard semantics of the STEP information model.

However, in origin and purpose, the STEP PDM Schema and the OMG PDM Enablers were not defined to be necessarily dependent on each other. Each can be used and are used in environments that do not include the other.

For example, STEP files can be used to exchange PDM data between different enterprises that do not have the need for run-time interactions between systems.

Conversely, the PDM Enablers interfaces can be used in an environment in which the data elements and constraints do not conform to the STEP standard. This is effective as long as the client and the server have a common understanding of the names and meanings of properties and relationships according to the information schema of the enterprise. And, of course, even in this environment, the STEP standard is useful used for information exchange to and from other enterprises.

4.3 Semantic comparison between the two Standards

This section describes the relationships between the two standards in terms of important PDM capabilities. A separate document addresses in detail the technical aspects of the two standards and mapping specifications between the two standards.

Any comparison between the models does not compare "like with like". There are differences in the level of details and overall abstraction mechanisms used in developing each of the models that are a result of the different implementation paradigms for which they were designed.

The OMG PDM Enablers provide an interface object model useful for operating on distributed objects whose actual implementation is internal to the PDM system. The STEP PDM schema has no formal concept of operations; it describes static PDM information as entities and attributes that will have standard exchange representations. This means that a direct object-to-object comparison of the models inevitably leads to some discrepancy. Some common concepts are modeled completely differently; and some objects that exist in one model have no comparable counterpart in the other model.

Table 2 - Relationships between STEP PDM Schema 1.1 and OMG PDM Enablers Modules, shows the correspondence between the major modules of the two standards. The detailed mapping of entities, attributes and operations between the two models is the subject of a separate paper. In this section we describe only the major PDM concepts and capabilities that are supported by the two standards, and the important commonalities and differences in the concepts they support.

4.3.1 Part Structure and Part Properties

Management of part structure information – the make-up of an assembly – and part definition information is the principal role of a PDM system. In this area, both standards address essentially the same concerns, but the models are somewhat different. There are several different concerns:

Approach to defining part properties

The STEP approach to defining part properties is to develop detailed models of properties, property elements and relationships, organized around particular types of products and particular views of products. The STEP PDM Schema is designed to work with these detailed property models as expansions of **product definition** objects, but it concentrates on the labeling and management of parts, part structures and part models, which is the traditional domain of PDM systems. And in managing these models, product definition objects are labeled with specific purpose and content attributes.

The OMG PDM Enablers sees detailed property models as entirely out of scope and concerns itself *only* with the labeling and management of parts, part structures and part models. It divides part property information into three categories, according to how the information is managed, rather than what it is. Part Documents are descriptions of parts managed as Document/File objects that are known to be intelligible to some other systems, but not to the PDM itself. (The PDM Schema equivalent is a product definition with associated documents.) Part Structures are descriptions of parts in terms of their composition from other parts. (The PDM Schema equivalent is Usage relationships between product definitions.) Part Data objects are "enterprise-specific" collections of simple information units, called Attributes, which are explicitly entered into the PDM schema in a particular installation. Attributes can be anything the PDM system supports and the local schema describes, but there is no way to represent complex relationships among these information units. Thus these are primitive "product definition" objects that can be used by applications interfacing directly with the PDM, but they are not intended to be equivalent to STEP detailed property models. And all of these Part Document, Part Structure and Part Data objects can be "qualified" by information that identifies their purpose and content.

In large measure, these two approaches can carry the same information relative to PDM capabilities, but the models are different, and some of the relationships are not obvious. And in particular, a number of the explicit part attributes in the PDM Schema are supported in the PDM Enablers via the Attributable feature of Part Data, Part Document and Part Structure objects.

Versioning

The PdmProductStructureDefinition Enabler models part definition data using a three-tier approach to version control. It defines operations on a PartMaster, which represents all part models conforming to a common form/fit/function specification, a PartRevision, which represents a formal change-level for the purpose of related engineering management services, such as notification and signoff, and PartDataIterations, which represent versions of specifications controlled (informally) by the engineering staff. The PDM Enablers must support the full multidiscipline life cycle of a product – a product/process definition in the course of design, review, rework, approval, release and revision. As concurrent engineering practices are deployed, it is important to support the independent evolution of different aspects of a part description (geometry, schematics, analysis, etc.). The PDM Enablers specification, therefore, supports management of simultaneous informal versions of different elements of a product description, in order to accommodate various engineering management practices.

The STEP PDM Schema has a two-tier approach to version control, defining a **product** object, which represents the common reference for form, fit and function, and a **product-definition-formation** (product version) object, to which all of the definition objects are attached. The naming and relationships among the versions define the formal and informal characteristics of particular versions. The primary purpose of the PDM schema is to support controlled releases and formal exchanges between trading partners, and it defines a number of specific attributes and relationships to support formal versioning and version labeling, but the same objects will support informal engineering exchanges. The STEP PDM Schema provides separate **product-definition** objects to carry separate specifications for the part for different views, aspects and disciplines, but it does not explicitly provide for separate versioning of those specifications. But this is because it is unlikely that systems would want to *exchange* multiple engineering versions of a particular discipline specification at one time.

So there is a difference in the formal model of versioning, but both models can support the same information and a mapping between these two approaches can be specified. The difference relates to the difference between exchanges of PDM information and access to PDM information management services.

Assembly Structure

The STEP PDM Schema model requires a specific revision of an assembly specification to refer to specific revisions of the component parts. This is a convenient representation for formal exchanges, representing the proper combination of specifications for assembly makeup at a release point. But this structure model is less suitable for capturing the frequent updates from different engineering organizations during product development, as each change in the structure would need to be captured. The PDM Enablers allow for many different iterations of the same assembly structure and requires a specific revision of an assembly specification to point to a component Part Master. The assembly structure specification is not changed unless the profile of parts (PartMasters) that constitute the assembly changes.

This difference in model capabilities is a consequence of the distinction between managing multiple specifications during the engineering process (a PDM Enablers concern) and making a proper selection of specifications for delivery of a complete product specification to archive or to some other system (the PDM Schema concern).

Digital Mockup information

The STEP PDM Schema provides for exchange of specific information objects (shape and position) to support digital mockup of a mechanical assembly. The PDM Enablers (v1.1) has no such objects. While such information can be supported by its user-defined attribution capabilities (Attributable operations), standardization of those functions must wait for a new revision of the PDM Enablers.

4.3.2 Document Management

The OMG PdmDocumentManagement Enabler defines Document objects and provides services to support management of electronic documents and non-electronic documents. These include support for operational behavior, such as during check-in and check-out, and storage in electronic vaults. Furthermore, the Enabler defines Document Revision and Iteration objects and relationships analogous to those for part versions. It associates files with Documents, and it allows user-defined relationships on these files, but it does not explicitly address document structure.

The STEP PDM schema defines logical Documents and their association to files as information sets, without file management operations and concerns. The PDM Schema does not explicitly address Document revisions or structure, but it provides a capability for representing a Document as a **product** object, and using associated product description capabilities to describe versions and structure.

The capabilities of the two systems for representing Documents are very comparable, but both require development of some particular conventions for supporting specific document management strategies. In the STEP view, these conventions are addressed in specific Application Protocols, such as AP232. In the OMG view, these conventions are created in the "enterprise-specific" internal schema for the PDM system. When the enterprise-specific schema matches a STEP AP, the two standards can be used consistently.

4.3.3 Identification Concepts

Both standards support multiple identifications of the principal PDM-managed objects -- Parts and Documents -- so that a given object can have different names and numbers for different reference contexts. E.g. a Part can have our part-number and also the customer's part-number and the supplier's part-number, and two systems of revision-labeling can be used for the same documents. Both standards also support short text descriptions of objects that are distinct from their identifiers. The PDM Schema distinguishes a primary identifier (the identifier most important to the particular exchange) from "aliases". The PDM Enablers simply provide for multiple identifiers, each having its own context, with the assumption that users will define a "primary context" for retrievals, and different users may define different ones.

4.3.4 Authorization

The PDM Enablers consider Persons, Organizations, and Programs as specific cases of a more general concept of Actor, which is itself defined in terms of it's ability to initiate actions on other objects. This notion is important to supporting "access control" and workflow management, both of which are common features of PDMs that are largely outside the scope of both standards. The STEP PDM Schema considers person and program separately and independent of their effect on a system interface.

The STEP PDM Schema provides explicit information units for describing the Certification or Approval of PDM objects. The OMG PDM Enablers does not provide these. Certification information can be supported via the Attributable capabilities mentioned above, but that mechanism can not properly support Approval chains.

The models for attachment of Security Classifications to Parts, Documents and other managed objects are intentionally identical.

4.3.5 Effectivity and Configuration

Both the PDM Enablers and the PDM Schema allow part descriptions, assembly/component usage relationships, and other important relationships to be controlled by effectivity specifications – identifying dates, serial/lot numbers, or factory locations for which particular specifications are valid. The PDM Enablers also allow such relationships to be labeled or controlled by lifecycle and discipline *views*. The PDM Schema allows specifications to carry explicit life-cycle context and discipline labels, thereby providing equivalent capability for parts and documents.

4.3.6 Change Management

The PDM Enablers identifies four processes, issue collection, requesting change, implementing change and notification of change. These processes have associations to the person initiating them where relevant and all are associated are to the items subject to change and the persons affected by the change. The PDM enablers allow for items that are formal revisions (documents, parts etc) to be subject to change but iterations of revisions are not associated to change. So through the informal versioning capability, the PDM enablers are able to capture both formal and informal change of PDM information. The STEP PDM schema describes the formal request for change, the change order that is used to implement the request and the description of some of the work undertaken to fulfil the change order. The PDM schema does take a static view of some activities that have taken place such as a result of the change order so that the Release State of the data being exchanged is captured too.

Broadly speaking both standards are similar in that they identify requests for change, implementing change and the relationship to the items subject to change. Also, both standards capture the inter-relation between all these concepts. The differences again relate to the difference between exchanges of PDM information and access to PDM information management services.

4.3.7 Part Classification

The STEP PDM Schema supports capture and exchange of part classification information. There is no explicit support for such information in the PDM Enablers, although the generic Attributable interface may be used to attach some classification information units to Part objects. Part classification is an area in which the needed services go well beyond the management of product data and are commonly provided by companion products to the PDM systems. And for this reason, part classification concerns are out of scope for the PDM Enablers.

4.3.8 Options Configuration Management

Options Management – specifications of product options, their interactions, and relationships to bills of materials – is a unit of STEP AP214 that is important to several industries. The PdmConfigurationManagement Enabler supports the AP214 model almost verbatim. This capability, however, was not included in the scope of any other STEP AP at the time of development of PDM Schema v1.1, and was therefore out of scope for that standard.

4.3.9 Manufacturing Implementation

Support for tooling and process specifications (routings, control programs, recipes and operator instructions) – is partly supported by a unit of STEP AP214 (Automotive Design). The PdmManufacturingImplementation Enabler provides support for a somewhat different model that was influenced by other manufacturing organizations and ERP vendors. . This capability, however, was not included in the scope of any other STEP AP at the time of development of PDM Schema v1.1, and was therefore out of scope for that standard.

4.3.10 Baseline

The PdmBaseline Enabler provides services that deal with identifying and managing a set of specifications as a conceptual unit, such as a product release version, or a package to be exchanged. A particular STEP data set conforming to the STEP PDM Schema refers implicitly to a “baseline”, in the sense that it is exactly the package of specifications to be exchanged as a unit. So this concept is implicitly supported, although technically out of scope of the STEP PDM Schema.

4.3.11 STEP

The PdmSTEP Enabler provides interface operations to export and import exchange files conforming to particular STEP Application Protocols (instances of the PDM Schema). This is an important companion standard that is itself out of scope of the PDM Schema.

5 Usage – Applying the Two Standards to Business Scenarios

The intent of this section is to describe a series of business scenarios along with a technical approach for addressing the requirements found within the scenario using the appropriate technology. In some cases this means using the OMG PDM Enablers. In some cases this means using the STEP PDM Schema. In some cases this means using both the OMG PDM Enablers and the STEP PDM Schema together. While it is true that the requirements from several of the scenarios could be met using either the PDM Enablers or the PDM Schema, care has been taken to describe the use of the technology deemed most appropriate by the authors who were involved in the creation of these two standards.

5.1 Release of a contracted design

Release of a contracted design requires reliable and secured data exchange procedures.

5.1.1 Description of Scenario

This scenario describes the relationship between a partner A designing a subsystem to be build by another partner B the main manufacturer. Partner A is working with the enterprise PDM system. At a certain stage of product development the design has to be released and approved to be sent out to the main manufacturer partner B. The release data set of the required product structure has to be identified, marked for release, extracted and prepared for the exchange process. The actual move of the data is organized with the help of established data packaging procedures ensuring a safe communication between identified and certified trusted data sender and data receiver. Incoming data must be checked into the enterprise PDM system of partner B. This is done clearly indicating the originator of the design data to be partner A outside of the enterprise of partner B. The communicated product structure will become a separable part of the receiver's product structure.

5.1.2 Technical Approach

The problem of data exchange between a contractor and a main manufacturer, at predefined stages of product development, is resolved by the following: a STEP PDM Schema output processor of the PDM system of partner B; an appropriate data exchange manager software to guarantee a safe communication; and a STEP PDM Schema input processor of the PDM system of partner A.

The output and the input processors have to be harmonized to communicate correct interpretable product data structures. The internal proprietary data models of the contractor PDM system and the main manufacturer PDM system need not be exposed. The mapping of the transferred information to the internal data model of the manufacturers PDM system is beyond the control of the data sender.

This is working practice and requires proper setup and preparation at the sender's and receiver's site. Typical requirements not addressed by the STEP PDM Schema approach alone are authentication of the sender, the receiver and the communicated documents, security aspects, legal aspects etc.

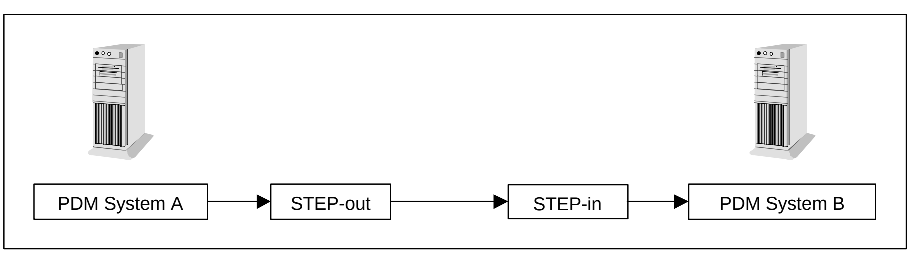


Figure 3 - Release of a contracted Design

5.2 PDM Enabled Applications

By providing PDM Enabled applications, vendors of software reduce the cost and increase the attractiveness of their software.

5.2.1 Description of Scenario

In this scenario, the vendor of a Computer Aided Design (CAD) system has many existing and potential customers with a wide variety of different commercially available and private PDM systems. These customers are asking for convenient and seamless integration between the CAD system and their PDM system.

Responding to this market need, the CAD vendor wishes to provide user interface and integration features that interoperate with a PDM system, such as:

- Create a new design document
- Check in a model file for the document.
- Adjust the engineering view of product structure
- Synchronize the values of attributes of the product between the CAD model and the PDM system

The vendor wants the CAD software to interact with as many different PDM systems as possible, but realizes that the cost of designing, implementing, and supporting software using the proprietary interfaces of each separate major PDM system is very expensive.

5.2.2 Technical Approach

A single design and implementation relying on the standard PDM Enablers interfaces interacts with most major PDM systems on the market. The PDM system vendors are responsible for supporting the PDM Enablers server.

Customers have an incentive to buy the PDM Enabled CAD application because it will be easy to incorporate it into their suite of other integrated applications that use the PDM Enablers standard.

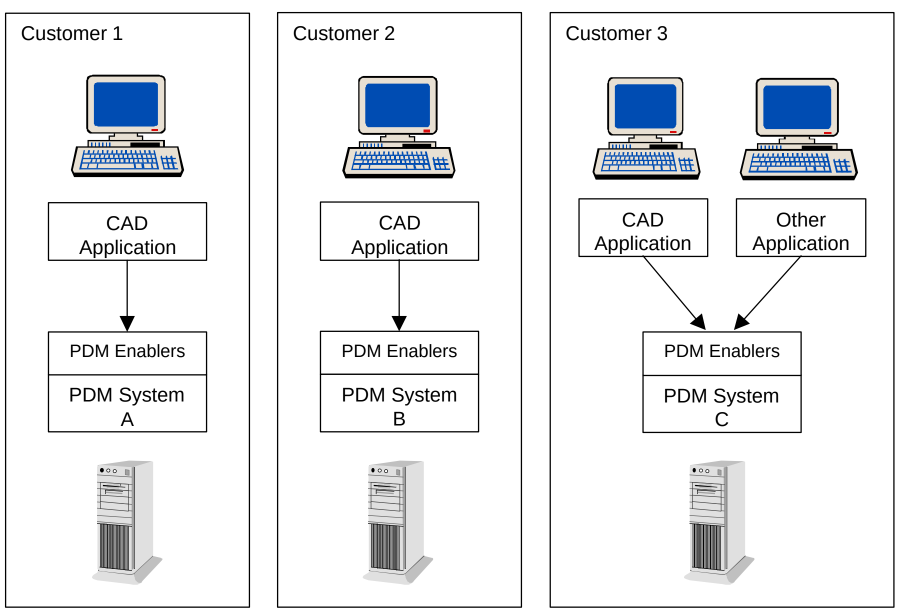


Figure 4 - PDM Enabled Applications

5.3 Product Structure Viewer

The PDM Enablers specification allows the design and implementation of a single client program that can be used with a variety of different PDM system servers.

5.3.1 Description of Scenario

In this scenario, a company wishes to supply its employees a product structure viewer, which is capable of showing product structures and analyzing them for conformance to certain unique business policies. The company has a diverse set of four different enterprise and workgroup PDM systems, in operation at different sites. One of these is a legacy proprietary system that will be probably be phased out in favor of a commercially available system over the next few years. Each of the PDM systems has a PDM Enablers interface.

5.3.2 Technical Approach

Rather than writing a custom client for each of the four separate systems, the IS staff writes a single client which uses the PDM Enablers interfaces. They prepare and deploy a single application program, user documentation, and training materials. When the legacy system is replaced by another PDM system, the viewer operates without change on the new PDM system.

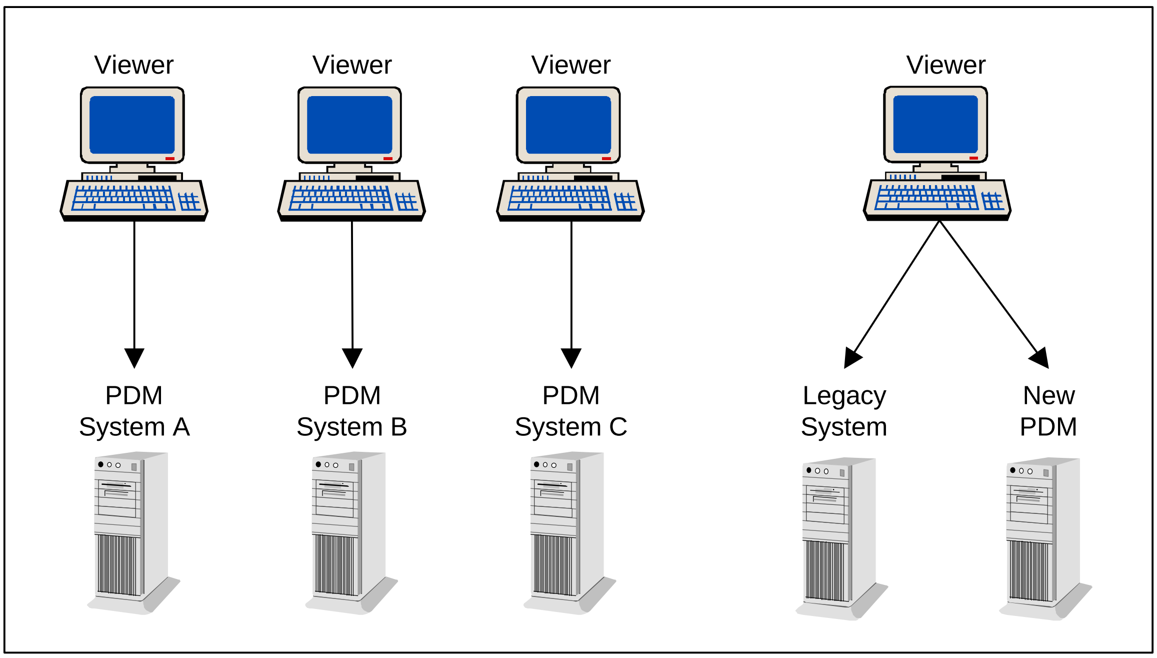


Figure 5 - Product Structure Viewer

5.4 Gradual Migration to new PDM Systems

The file exchange type of scenario is typified by some large volume of information to be moved (or shared in a disconnected manner) between N partners. This type of interchange allows for scrutiny of the incoming information prior to its introduction into the production environment.

5.4.1 Description of Scenario

In this scenario, company A is contemplating replacing their enterprise or workgroup PDM system with a newer vintage. The IS group and users have decided that this must be done in a 3 step process which will: load the new system with data, run the new system in parallel with the current system (during tailoring, tuning and user training), and sunset the current system.

In order to reduce the impact of steps 1 and 2, both systems (current and future) will be fitted with translator or adapters to exchange the initial information load data and the parallel production information. Company A has decided it is far too risky to ask the users to enter the parallel production information in both systems.

5.4.2 Technical Approach

Periodic SynchronizationThe STEP PDM schema view of this scenario would appear as in Figure 6 – File Exchange and the PDM Schema

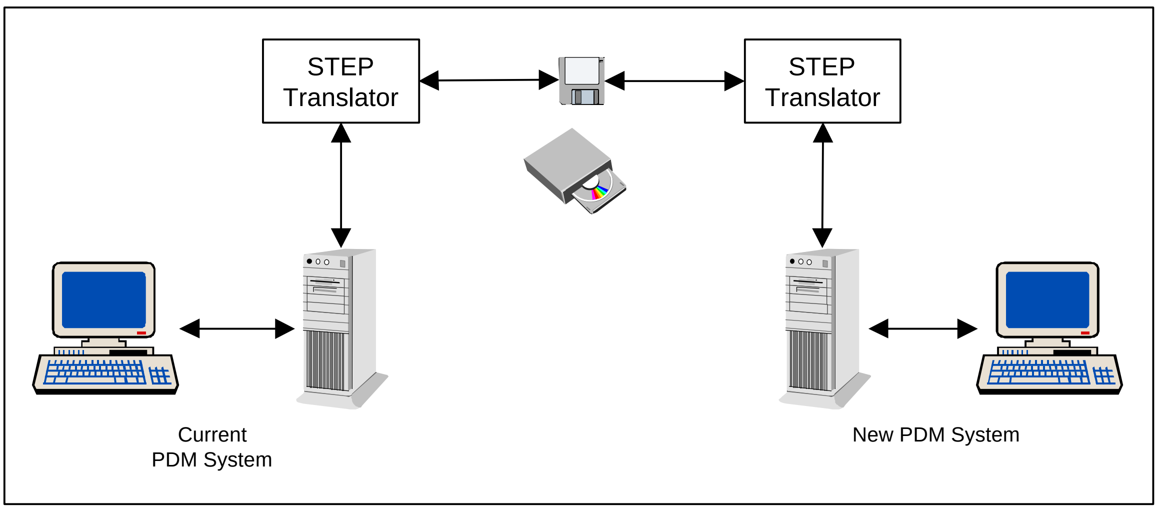


Figure 6 – File Exchange and the PDM Schema

The initial information load and parallel operation would be done via physical file transfer. The parallel operation (in this case) would send the data for the parts that change in update files. The receiver would replace the data or add it. This configuration makes sense if the parallel operation can be done in a batch over night paradigm. If the parallel operation must be concurrent, STEP should not be used for the parallel operation, as this would be better done with the PDM Enablers.

5.5 Concurrent Parallel Operation

The combined view of this scenario would appear as in Figure 7 - Concurrent Parallel Operation. The concurrent parallel operation scenario is typified by some large volume of information to be moved in a one-time operation (bulk file load) which is followed by cross system updates on a real time basis.

5.5.1 Description of Scenario

In this scenario, company A is contemplating replacing their enterprise or workgroup PDM system with a newer vintage. The IS group and users have decided that this must be done in a 3 step process which will: load the new system with data, run the new system in parallel with the current system (during tailoring, tuning and user training), and sunset the current system.

In order to reduce the impact of steps 1 and 2, both systems (current and future) will be fitted with translators to exchange the initial information load data and adapters to exchange the parallel production information.

5.5.2 Technical Approach

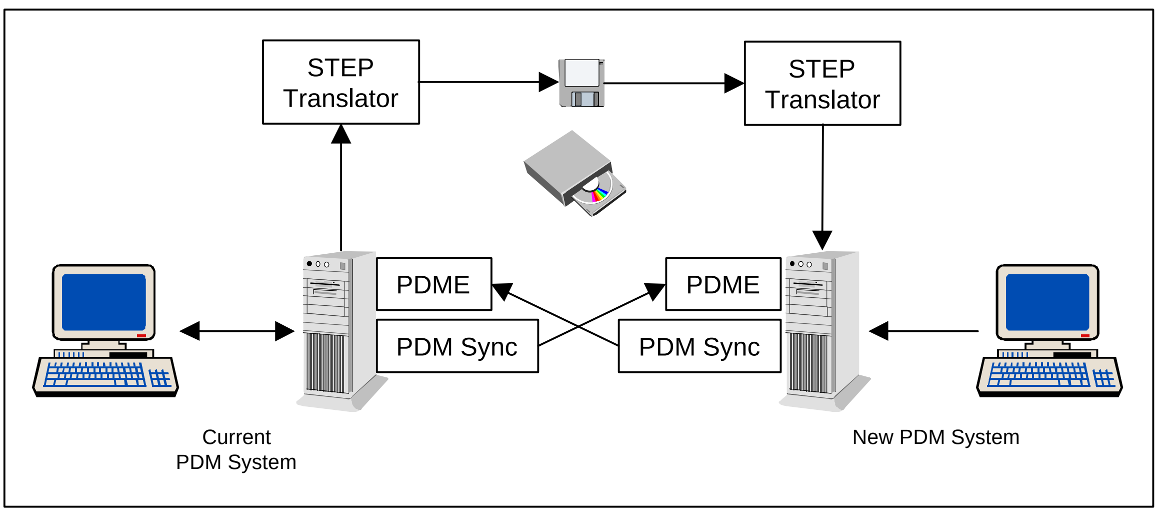


Figure 7 - Concurrent Parallel Operation

The initial information load would be done via STEP physical file transfer. The parallel operation would be done via the PDM enablers using transaction based net change.

This is configuration is the most efficient since each technology (STEP/PDM Enablers) is playing to its strength.

5.6 Digital Mockup / Collaborative Design

Digital Mock-up (DMU) technology is a 3-D model of the entire product with each of the components and assemblies placed appropriately in 3-D space. The DMU is used to communicate the evolution of a design among the designers who may be physically and organizationally separated. DMU's are often used to check for interference, fit and clearances between parts, engineering analyses, visualization, etc. The DMU consists of the geometric shape of the various components, placement and orientation information, the product structure as well as version and configuration control information.

5.6.1 Description of Scenario

In this scenario, the product is being designed by a team whose members span company or organization boundaries who need to collaborate on the product structure, and the product shape. The members of the team communicate via a Digital Mock-up (DMU). As the product design evolves, the various team members need to communicate changes and updates to the design with their teammates. The computing systems being used include CAD, Workflow or Technical Data Management (TDM), analysis applications and PDM systems. Depending on the relationship between organizations, there may or may not be a master repository for released engineering data.

Each organization that has design responsibility may manage their design (product structure and shape data) in their own systems. In order to coordinate the design across organizations, these systems need to share information. In many cases, the technology is not available to federate the various team members' systems together into a single, distributed system. For many complex geometric applications such as spatial analysis and visualization, the shape data needs to exist locally to provide acceptable performance. For these reasons, each member site typically has a copy of the entire design, the configuration of which is coordinated with the other sites. Each site need to be "in synch" with each other to ensure that everyone is working with the correct configuration.

The requirements for data exchange and sharing might the exchange of assemblies of geometric shapes between TDM or PDM systems including product identification and configuration information, comparing product configurations between sites, computing change deltas between systems

5.6.2 Technical Approach

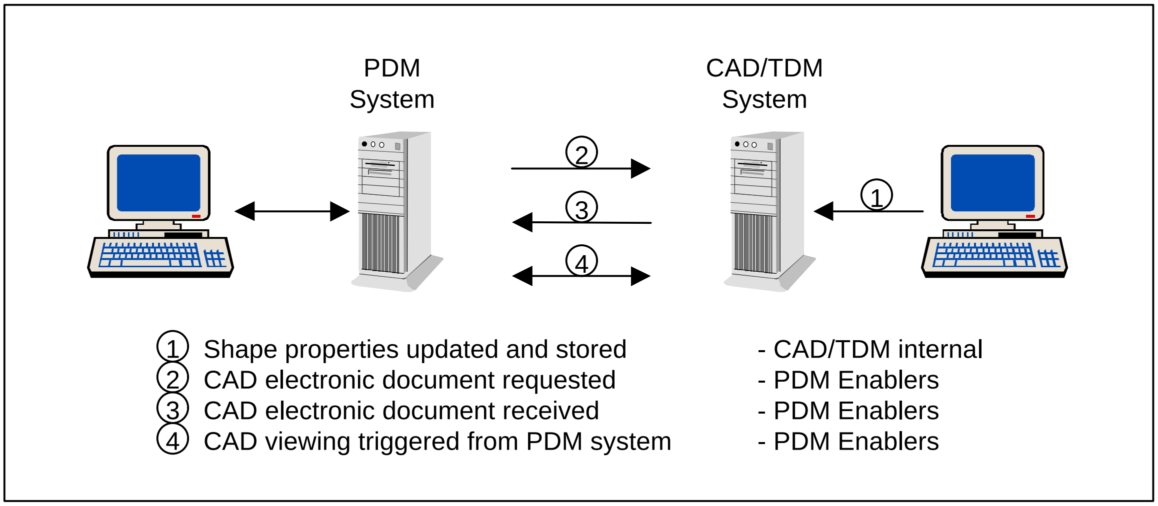


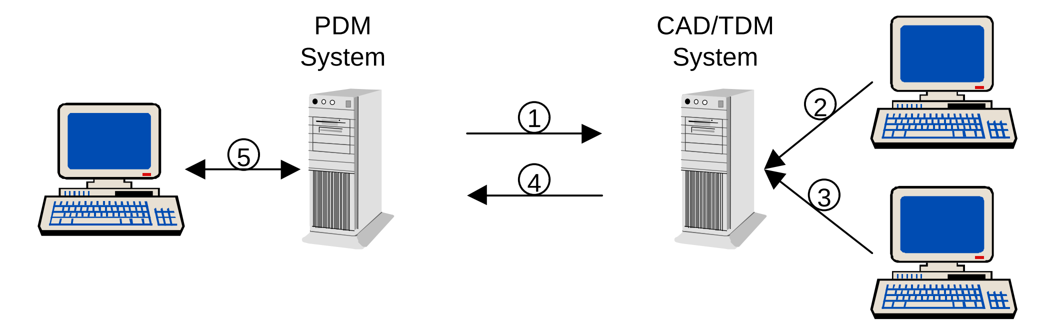
Figure 8 - Scenario providing solution using OMG PDM Enablers

In this scenario, two design teams, Team A and Team B, are collaborating on the design of a product.

(1) Team A is working on the digital mock-up of the product and is frequently updating the detailed design and associated product structure. The product structure of the digital mock-up is maintained by Team A's Technical Data Management system that manages the information for the CAD systems. Team B's Technical Data Management system compares its product structure with Team A's to ensure that both teams are working to the same configuration. The comparing application uses the PDM Enablers to ensure that the product structures in both TDM systems are the same.

(2) and (3) The structural designers are working on a related area of the product structure and wish to view the design associated with that area. The file location information of the appropriate CAD files is requested from the TDM system and list of documents associated with the relevant elements of the product structure is updated using the Document interface of the PDM enablers.

(4) The PDM enablers are then used to trigger an event to view the files across the network from the PDM system. The information model used by all systems behind the interface is not made public.

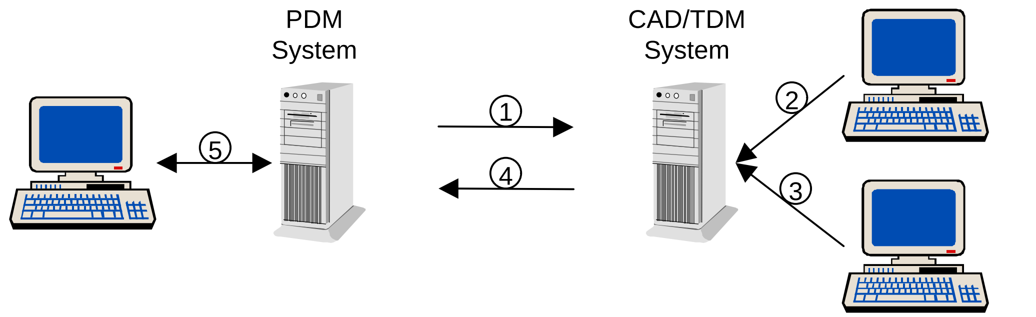


1 Product structure requested - STEP PDM SDAI
2 TDM structure updated and stored - CAD/TDM internal
3 Shape properties updated and stored - CAD/TDM internal
4 CAD shape representation and position requested - STEP PDM SDAI
5 Product structure positions updated and stored - PDM internal

Figure 9 - Scenario providing solution using STEP PDM Schema

(1) and (2) The latest product structure updates from Team A is extracted from the PDM system and converted into a STEP Part 21 file conforming to the PDM schema by the product shape team that as it happens use the TDM system to manage a heterogeneous CAD environment. The TDM system accesses the PDM system using SDAI and other appropriate APIs based on the STEP PDM schema. This type of access provides detailed information on the shape properties of individual elements of the product structure and standardizes the information being used at the attribute level.

(3) The STEP file is imported into Team B's PDM system and integrated into the existing product structure. Team B then requests the geometry files for the new or updated designs from Team A's TDM system. The resulting geometry files and then loaded into Team B's TDM and reconciled with the product structure in Team B's PDM system using SDAI calls.



1 Product structure requested - PDM Enablers and STEP PDM SDAI
2 TDM structure updated and stored - CAD/TDM internal
3 Shape properties updated and stored - CAD/TDM internal
4 CAD shape representation and position requested - PDM Enablers and STEP PDM SDAI
5 Product structure positions updated and stored - PDM internal

Figure 10 - Scenario providing solution using STEP PDM SDAI and PDM Enablers

(1) A product structure comparison application is executed by Team B. The application uses the PDM Enablers to navigate the product structure of Team A's DPA database and compares it to the product structure in Team B's DPA database. Where it finds a difference in configuration, the newer product structure data is requested from Team A's PDM system. The detailed product data is accessed via SDAI calls based on the PDM schema

(2) The product structure for the new configuration is evaluated and the geometric definition for any new or updated parts are requested from Team A's TDM system. The data is sent as one or more STEP Part 21 files.

5.7 *Federated PDM*

The Federated PDM enables collaboration and information sharing between internal organizations, which may or may not be widely dispersed, teaming partners, suppliers, and customers. The federated approach allows access to product information created by any and all groups that participate in the federated environment, while maintaining local autonomy and control of databases (single source of Product Data). Information is shared in a virtual PDM environment.

5.7.1 Description of Scenario

In this scenario the Tooling and Assembly Detail is created in the Product Center utilizing information from different PDM/CAD Systems. The User must be able to Create, Retrieve, Update and Delete the information (Part Package Data) associated with Design of the Parts. The User invokes operations to Create, Retrieve, Update or Delete the information related to the Part from a client, where the client may be a front end PDM managing the requests. The data is retrieved and any updates are written back to underlying datastores (multiple systems (federated)) without the User knowing where the actual datastores may reside.

5.7.2 Technical Approach

Development of Standards to accomplish this technology (Federated PDM) is currently in work with the Standards Organizations through RFI/RFP submissions.

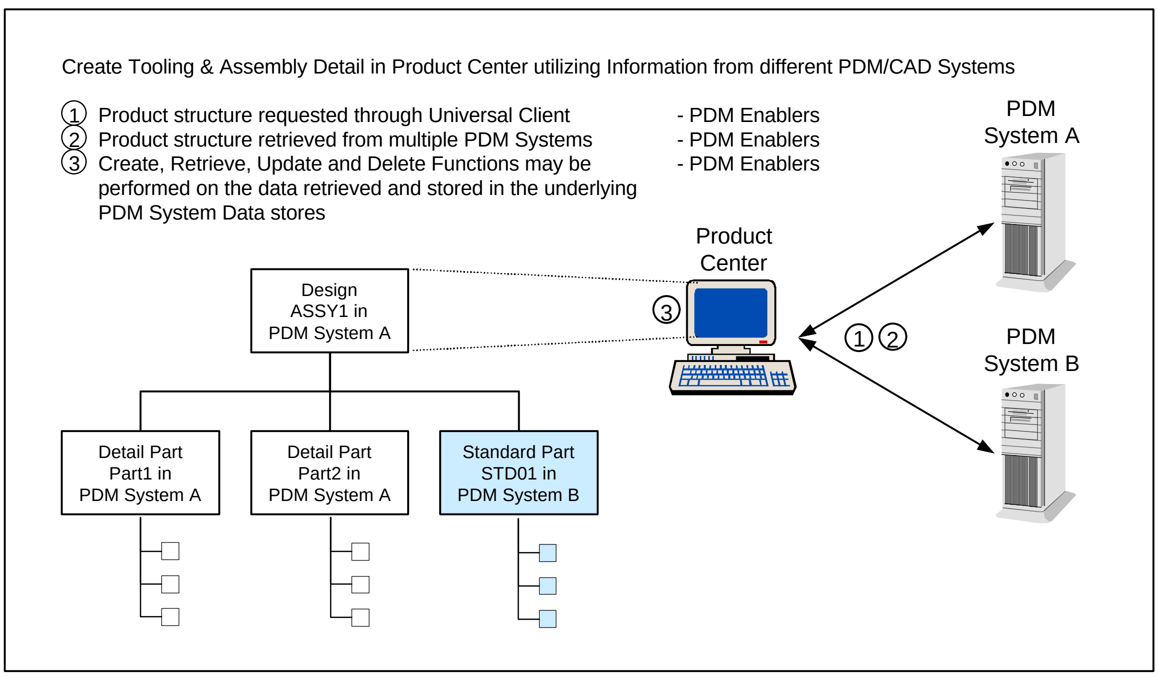


Figure 11 - Federated PDM

Product Center User requests and reviews information for Assembly ASSY1 from Universal Client (may be front-end PDM system).

Product Center PDM (or Universal Client), through CORBA and PDM Enabler interfaces, accesses and retrieves needed information from various PDMs and Datastores.

Product Center Universal Client displays Product Package (Structure, Notes, … BOM, Parts list, Geometry, and Associated Documentation and Visualization*)* information for each Part. In this example the Standard Part STD01 must be located in the "Standard Parts Database" PDM_2 and presented back to the Universal Client as a component of ASSY1, with it's associated information.

The Product Center User views ASSY1 Part Attributes (using Universal Client) and changes/modifies Detail PART1 Envelope LENGTH to support Manufacturing Center PROCESS Requirement.

When the User completes the modification, the updates are committed to the underlying datastores through PDM Enablers interfaces without the User knowing where the underlying datastores are located. Processing must accomplish attaching the reference from the revised Detail PART1 to ASSY1 Part.

Expanding on this example…

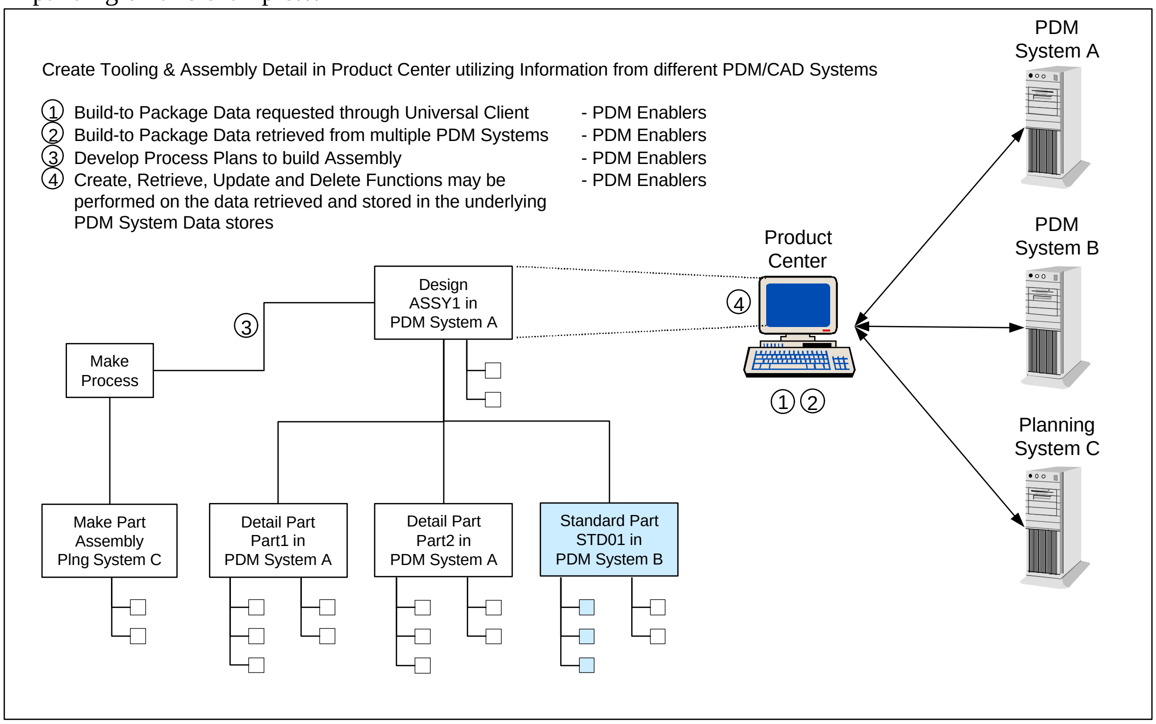


Figure 12 - Federation, Build-to Package

Product Center User requests and reviews information for Assembly ASSY1 from Universal Client.

Product Center Universal Client, through CORBA and PDM Enabler interfaces, accesses and retrieves needed information from various PDMs and Datastores.

Product Center Universal Client displays Product Package (Structure, Notes, … BOM, Parts list, Geometry, and Associated Documentation and Visualization*)* information for each Part. In this example the Standard Part STD01 must be located in the "Standard Parts Database" PDM_2 and presented back to the Universal Client as a component of ASSY1, with it's associated information. Process Plans will be created and associated to the Product Package and Stored in Manufacturing Process Planning Systems.

Various Views are Setup in the Product Center Universal Client from Data in the Other PDM / CAD and Database Systems.

Product Center User adds Information (Manufacturing plans, Graphic aids, Tools, Standard parts (therefore a change structure), and NC programs) to Assembly ASSY1, DETAIL PART 1, and DETAIL PART 2.

The Product Center User views ASSY1 Part Attributes (using Universal Client) and changes/modifies Detail PART1 Envelope LENGTH to support Manufacturing Center PROCESS Requirement.

Product Center adds MAKE PROCESS INFORMATION to Product Package Tools, and assembles the Assembly ASSY1.

When the User completes the modification, the updates are committed to the underlying datastores through PDM Enablers interfaces without the User knowing where the underlying datastores are located. Processing must accomplish attaching the reference from the revised Detail PART1 to ASSY1 Part, as well as the MAKE PROCESS and Make Part Assembly references to ASSY1.

6 Bibliography – References to Related Specifications and Standards

ISO 10303-21, *Industrial automation systems and integration - Product data representation and exchange - Part 21 : Implementation method: Clear text encoding of the exchange structure*.

ISO 10303-22, *Industrial automation systems and integration - Product data representation and exchange - Part 22 : Implementation method: Standard data access interface specification*.

ISO 10303-203, *Industrial automation systems and integration - Product data representation and exchange - Part 203 : Application protocol: Configuration controlled design*.

ISO 10303-210, *Industrial automation systems and integration - Product data representation and exchange - Part 214 : Application protocol: Electronic assembly, interconnet, and packaging design*.

ISO 10303-212, *Industrial automation systems and integration - Product data representation and exchange - Part 214 : Application protocol: Electrotechnical design and installation*.

ISO 10303-214, *Industrial automation systems and integration - Product data representation and exchange - Part 214 : Application protocol: Core data for automotive mechanical design processes*.

ISO 10303-232, *Industrial automation systems and integration - Product data representation and exchange - Part 214 : Application protocol: Technical data packaging core information and exchange*.

ISO TC184/SC4. *The Complete List of STEP Parts*, http://www.nist.gov/editing/step/titles/

The PDM Implementors Forum. *The STEP PDM Schema Version 1.1*, http://WWW.PDM-IF.ORG/pdm_schema/

ISO/IEC JTC 1/SC21, *Open Distributed Processing – Reference Model – Part 1: Overview; International Standard 10746-1, ITU-T Recommendation X.901*

The Object Management Group, *A Discussion of the Object Management Architecture*, http://www.omg.org/library/oma2.html

The Object Management Group, *CORBA / IIOP 2.2 Specification*, http://www.omg.org/library/c2indx.html

The Object Management Group, *CORBAservices Specification*, http://www.omg.org/library/csindx.html

OMG Manufacturing Domain Task Force, *Manufacturing Enterprise Systems, Whitepaper*, 9 February 1996, http://www.omg.org/cgi-bin/doc?mfg/96-01-02

OMG Manufacturing Domain Task Force, *Product Data Management Enablers RFP: mfg/96-08-01*, http://www.omg.org/cgi-bin/doc?mfg/96-08-01

OMG Manufacturing Domain Task Force, *Product Data Management Enablers Version 1.0 Feb 1998*, Joint Revised Submission mfg/98-01-01, http://www.omg.org/cgi-bin/doc?mfg/98-01-01

OMG Manufacturing Domain Task Force, *Product Data Management Enablers Version 1.0 Errata mfg/98-02-01*, http://www.omg.org/cgi-bin/doc?mfg/98-02-01

OMG Manufacturing Domain Task Force, *Product Data Management Enablers Version 1.0 Convenience Document: mfg/98-02-02*, http://www.omg.org/cgi-bin/doc?mfg/98-02-02

OMG Product Data Management Enablers V1.1 Revision Task Force, *Final Report, March 1999, dtc/99-02-01* http://www.omg.org/cgi-bin/doc?dtc/99-02-01

7 Appendices

7.1 Appendix A

STEP and OMG Product Data Management Specifications - A Guide for Software Developers (Note: This Document is currently in work by the STEP and OMG Communities as part the STEP / OMG Harmonization Effort, and includes mapping and technical analysis between the two Standards)